Automated Readability Determination of Text Documents using Shallow and Syntactic Features

I. Mandal[1], M. A. Sultan[2], F.A. Hoque[3] and M. S. Ahmed[4]
[1] Department of CSE, University of Information Technology & Sciences (UITS), Dhaka, Bangladesh
[2]Department of Computer Science & Engineering, University of Dhaka, Dhaka-1000, Bangladesh
[3] Department of CSE, University of Information Technology & Sciences (UITS), Dhaka, Bangladesh
[4] Department of CSE, University of Information Technology & Sciences (UITS), Dhaka, Bangladesh

Abstract - **Readability assessment of a text document is an important research topic. Readability assessment of a document is the process of measuring the difficulty level of that document. Traditionally research on text readability assessment has relied on shallow document features like average sentence length or the difficulty of the vocabulary or average syllable count. But recent advances in computational language processing and the advent of machine learning algorithms have made the task of readability assessment automatable by extracting various other document features. In this study, for readability assessment we have used shallow and syntactic document features. Since they are sentence level features the extraction is less time-consuming compared to other families of features like discourse. The aim of this research is to increase the accuracy of readability assessment using shallow and syntactic features. Experiments on different input sets shows improvements over existing results found by changing the feature set for training classifiers.**

Key words: *Accuracy, machine learning, readability, shallow feature, syntactic feature, precision.*

1. Introduction

A reading difficulty measure can be described as a function that map a text to a numeric value according to its difficulty level or grade level. Input of these function are several features and the output is one of a set of grade level. In prior when readability was calculated by human some shallow features were considered. Using these shallow features we can easily compute readability considering any type of programming language. But using these shallow features, we cannot get better accuracy. Now the advancement of computational language processing and machine learning algorithms, it is possible to make readability assessment auto-mated. A document typically consists of many components such as text, image, and hyperlink etc. A very important subsystem of a document readability assessment tool is one that assesses the difficulty level of its text contents. Now a day, text documents are available in digital form in digital library or web pages. When someone wants to choose a document from a digital library or web source on a specific topic or when teachers suggest appropriate reading material for their students, an automated system for calculating readability level of a text will help the resource selection process to a large extent. But assessment readability of any text document time is an important factor. If it takes so much time to determine the readability of a text, then it is not expected.

So there must have a system for estimating readability in less time. An automated system for determining readability level saves the time and effort required in a manual assessment system. With the introduction of digital information repositories, automated text difficulty assessment system are thus of great importance.

The aim of this research work is to determine the readability level of a text document. Many features have been found to affect text readability. These include traditional shallow features (such as syllable per word and average sentence length etc), syntactic features (such as average number of verb phrases and average number of noun phrases per sentence and average number of SBAR per sentence etc), discourse features (such as entity-density features and entity grid features or lexical feature). We have selected our research area in shallow features and syntactic features. In this study, we incorporate these features with one another to get better accuracy. By using several machine learning algorithms such as Naive Base, F. Sebastiani (2002) and K. Collins-Thompson et al.(2004), support vector machines, Multilayer Perceptron, A. Hart(1992), Decision Tree, H. Isozaki (2001) and M. Bevilacqua et al.(2003), k-nearest neighbors' algorithm etc , we have measured the accuracy level. Performances of several classifiers in readability assessment with a combined set of shallow features and syntactic features have also evaluated. The experimental results on input data sets show the superiority of the proposed method in terms of accuracy and performance.

2. Types of features

Readability of a document depends on the writing style of the document and reading ability of readers. If one person has capability to understand a grade one document and requires understanding the grade four documents, then it is

Manuscript received May 25, 2010.
Manuscript revised July 15 , 2010.

too much difficult. In that case, if a documents readability level is known to all, then some-one can easily select the suitable document. So it is very important to have an automated system for calculating grade level. For determining readability, we have considered shallow and syntactic features.

Shallow Features: Document size is an important feature for determining grade level. For grade one, it is seen that total number of words in a document is between 100. Total number of words in grade five is between 350 to 550. So, using these features only or incorporate with another features, we can predict readability level. Total number of sentences is related to document size. For grade one, number of sentences is between 20 and 30 and for grade five documents, total number of sentences is between 70 and 90. But some times for grade one document, each sentence length is small but total number of sentences is large and for high grade document, number of sentences is minimum but sentence length is large. Considering these situations, we can get better accuracy if we count average sentence length. In shallow considering, used word in the document is another important factor, M. Heilman et al. (2008) and J. Chae, A. Nenkova (2009). Vocabulary level for a lower grade students are not so high comparing higher grade student. We have a frequently used vocabulary list. Considering these vocabulary list, we can easily determine grade level. In vocabulary list, there are 1000 words that are frequently used. But a word is used in documents in various forms such as in present, past or continuous form etc. To overcome this problem, steaming algorithm is used. After steaming each word, we can easily match the word from vocabulary list and document. Syllable is another important factor. Since in grade one level easy words have used, so number of monosyllables or disyllables are many. In higher grade level, polysyllables are frequently used. Combining the no. of monosyllable, disyllable and polysyllable with total syllable or average number of syllable predicts readability level. Some shallow formulas like Flesch-Kincaid formula, D. D'Alessandro et al.(2001), Gunning fog index, L. Feng et al. (2010) etc can be used to calculate the grade level. Using these formulas and combining these with another, we can easily calculate readability level. In corporate those features with one another we can get better result than combining all of those features or only used those features.

Syntactic Features: Syntactic features are those features in which grammatical condition are considered. A parsed tree have generated from each sentence. From this parsed tree, we can easily determine noun phrase, verb phrase, proper noun, adjective, SBAR etc. Using this, readability of a text is easily determined. We have used Stanford parser for parsed tree generation. Using Stanford parser we have generated parse tree of each sentence of document. Then using java programming language, we have collected our required information such as noun phrase, verb phrase etc. Using those features, we have extracted required syntactic features. In syntactic feature, I have considered average number. of noun phrase, average number of verb phrase, average number of SBAR and average tree height. Incorporate those features with one another and with other shallow features, we can get better output.

3. Method

A document may contain many features such as text, image, URL etc. This research work is focused to calculate the readability level of any text document automatically. To calculate readability we have to consider two factors. The first one is to form a set of features and then evaluate the performance of several classifiers in readability assessment using those features. For these purpose we have to train the classifier. For implementation of supervised machine learning algorithm WEKA workbench has been used. Using these features extracted from our document, we have applied our required classification algorithms like Naive Base algorithm, F. Sebastiani (2002) and K. Collins-Thompson et al.(2004)], Lib SVM, SMO, Multilayer Perceptron, A. Hart(1992)and decision tree algorithm, H. Isozaki (2001) and M. Bevilacqua et al.(2003). The steps of the method can be summarized as follows:

Corpus Selection: For training classification algorithm we need to have some annotated corpus. Annotated corpus is a corpus which difficulty level is known to us. For annotated corpus we select some reading comprehension for each grade level. We have collected some document or some reading comprehension in which grade level is given according to that document. We have extracted information from those documents. Here we consider five grade levels.

Feature Extraction: After selecting some document we have to extract feature. Shallow features are extracted using java programming language. Such that from a document total number of word, average numbers of sentences, syllable counts have calculated. In extracting syntactic features, we have to generate parse tree for each sentence. Then we calculate average noun phrase, average verb phrase, average SBAR, average parsed tree height etc. These values are put in raff file. Then using these features, several machine learning algorithm are used to classify a document.

Classification: Taking information from feature extraction we have classified our document using WEKA. For this perspective we used several machine learning algorithm such as Multilayer Perceptron, A. Hart (1992), Lib SVM etc. We have trained WEKA using those extracted information. Then apply 10 fold cross validation for finding accuracy level. After training WEKA using those data sets, if we give WEKA a document which readability level is unknown to us, it classifies the document.

4. Integration of New Features

In this study, shallow and syntactic features are considered. For feature extraction we have use many average value such as average sentence length, average noun phrase, average verb phrase, average tree height etc. Using these single features or combining these with other features we have calculated accuracy using several machine learning algorithms. The experiments showed that if we use frequency distribution of these value incorporate with average value, accuracy level change radically. We get much accuracy if average value and combining these values with its range value.

5. Experimental Results

For the purpose of analyzing and demonstrating the efficiency and effectiveness of the proposed method, we conducted some experiments at the computer laboratory of the Department of Computer Science and Engineering, University of Dhaka. Since the goal is to predict the complexity of any text document, we have some text documents as annotated corpus which difficulty level is predetermined. We have extracted features from this corpus. Then using 10 fold cross validation formula and several machine learning algorithms we determine the accuracy, precision and F-measure. All the experiments are performed on a Pentium IV 2.0GHz CPU with 4 GB of memory and running Windows XP. For different input sets, we had applied different machine learning algorithms and got different results. We also compared the result with previous work. It demonstrates that the proposed method possesses better performance than existing one.

5.1 Accuracy of Classification for Different Feature Sets:

In this research we had extracted many features of shallow and syntactic features. Accuracy achievement of those features is not the best result. But if we incorporate some feature with one another, we can get better accuracy result. We have WEKA workbench. Using these tools, we can test accuracy of several machine learning algorithms. The

5.1.1 Only Shallow Features

Set1: {Only considering total document size}.

Table 1: Set 1

Machine Learning	Accuracy	Precision	Recall	Fmeasure
Naïve Bayes	66.279%	0.644	0.663	0.639
LibSVM	51.163%	0.624	0.512	0.514
MultiLayerPerceptron	68.605%	0.662	0.686	0.656
SMO	53.488%	0.381	0.535	0.437
DecissionStump	46.512%	0.239	0.465	0.311

Set 2: {Average sentence length, total document size}

Table 2: Set 2

Machine Learning	Accuracy	Precision	Recall	Fmeasure
NaiveBayes	66.116%	0.652	0.651	0.646
LibSVM	46.511%	0.705	0.465	0.444
MultiLayerPerceptron	67.442%	0.677	0.674	0.672
SMO	54.651%	0.464	0.547	0.492
DecissionStump	46.511%	0.239	0.465	0.311

Set 3: {Average sentence length, frequency distribution of average sentence length, total document size}

Table 3: Set 3

Machine Learning	Accuracy	Precision	Recall	Fmeasure
NaiveBayes	58.140%	0.592	0.581	0.582
LibSVM	60.465%	0.715	0.605	0.607
MultiLayerPerceptron	68.605%	0.685	0.686	0.681
SMO	44.186%	0.411	0.442	0.397
DecissionStump	46.512%	0.239	0.465	0.311

Comparing the above two tables of 2 and 3, it is seen that accuracy is increased in different machine learning algorithm. If we use Neural Network (used Multilayer Perceptron) algorithm when only average sentence length is considered, we get 67% accuracy. But incorporate it with its frequency distribution, we get 69% accuracy. So we say that if we use average sentence length incorporate with its frequency distributions instead of using average sentence length only, we can get better accuracy.

Set 4: {Total syllable count}

Table 4: Set 4

Machine Learning	Accuracy	Precision	Recall	Fmeasure

NaiveBayes	39.535%	0.429	0.395	0.354
LibSVM	36.047%	0.66	0.36	0.281
MultiLayerPerceptron	58.40%	0.498	0.581	0.524
SMO	44.186%	0.411	0.442	0.397
DecissionStump	46.512%	0.239	0.465	0.311

Set 5: {Total syllable, average syllable per document}.

Table 5: Set 5

Machine Learning	Accuracy	Precision	Recall	Fmeasure
NaiveBayes	46.512%	0.458	0.465	0.446
LibSVM	36.047%	0.66	0.36	0.281
MultiLayerPerceptron	53.488%	0.454	0.535	0.481
SMO	44.186%	0.411	0.442	0.397
DecissionStump	46.512%	0.239	0.465	0.311

Comparing table 4 and table 5 it is seen that if we incorporate average syllable per word with total syllable, accuracy is increased. If we use NaiveBayes algorithm, when only total syllable count is considered we get 39% accuracy. But incorporate it with its average syllable per document, we get 46% accuracy. Again, if we compare table 5 and table 6 it is seen that accuracy increased when frequency distribution of average sentence length is used incorporate with average sentence length and total syllable. If we use MultiLayerPerceptron algorithm, when average sentence length and total syllable count is considered we get 53% accuracy. But incorporate it with its frequency distribution, we get 70% accuracy.

Set 6: {Total syllable, total document size, average sentence length, frequency distribution of average sentence length}

Table 6: Set 6

Machine Learning	Accuracy	Precision	Recall	Fmeasure
NaiveBayes	46.512%	0.458	0.465	0.446
LibSVM	34.884%	0.607	0.349	0.274
MultiLayerPerceptron	70.093%	0.721	0.721	0.716
SMO	53.488%	0.544	0.535	0.537
DecissionStump	46.512%	0.239	0.465	0.311

Set 7: {Average sentence length, probability of each sentence length, total document size}

Table 7: Set 7

Machine Learning	Accuracy	Precision	Recall	Fmeasure
NaiveBayes	51.163%	0.58	0.512	0. 512
LibSVM	36.047%	0.66	0.36	0.281
MultiLayerPerceptron	69.767%	0.681	0.698	0.685
SMO	52.326%	0.49	0.523	0.483
DecissionStump	46.512%	0.239	0.465	0.311

Set 8: {Average sentence length, probability of each sentence length, total document size, total syllable, average syllable}

Table 8: Set 8

Machine Learning	Accuracy	Precision	Recall	Fmeasure
NaiveBayes	52.326%	0.569	0.523	0.526
LibSVM	33.721%	0.449	0.337	0.237
MultiLayerPerceptron	70.930%	0.715	0.709	0.709
SMO	55.814%	0.514	0.558	0.52
DecissionStump	46.512%	0.239	0.465	0.311

Set 9: {Total syllable, average number of mono syllable, average number of disyllable, average number of trisyllable, average number of polysyllable, flesh-kincaid value}.

Table 9: Set 9

Machine Learning	Accuracy	Precision	Recall	Fmeasure
NaiveBayes	53.488%	0.531	0.535	0.529
LibSVM	41.861%	0.545	0.419	0.372
MultiLayerPerceptron	60.465%	0.613	0.605	0.608
SMO	51.163%	0.635	0.64	0.629
DecissionStump	46.512%	0.239	0.465	0.311

Set 10:{Total syllable, average number of mono syllable, average number of disyllable, average number of trisyllable, average number of polysyllable, average sentence length, frequency distribution of average sentence length ,flesh-kincaid value }.

Table 10: Set 10

Machine Learning	Accuracy	Precision	Recall	Fmeasure
NaiveBayes	53.488%	0.531	0.535	0.529
LibSVM	36.047%	0.66	0.36	0.281
MultiLayerPerceptron	67.442%	0.664	0.674	0.665
SMO	52.32%	0.479	0.523	0.48
DecissionStump	46.512%	0.239	0.465	0.311

Comparing table 7 and table 8 it is seen that accuracy is increased in different machine learning algorithms. If we use SMO algorithm, when average sentence length, probability of each sentence and total document size is considered we get 52% accuracy. But incorporate it with total syllable and average syllable, we get 55% accuracy.

If we compare table 9 and table 10 it is seen that accuracy is increased in each machine learning algorithm. If we use MultiLayerPerceptron algorithm, when average sentence length, total syllable and average number of mono, di, tri and polysyllable is considered we get 61% accuracy. But incorporate it with frequency distribution, we get 67% accuracy.

Set 11: {Frequently used word, total document size}

Table 11: Set 11

Machine Learning	Accuracy	Precision	Recall	Fmeasure
NaiveBayes	48.837%	0.545	0.488	0.5
LibSVM	67.442%	0.701	0.674	0.677
MultiLayerPerceptron	46.512%	0.48	0.465	0.47
SMO	50%	0.375	0.5	0.428
DecissionStump	46.512%	0.239	0.465	0.311

Set 12: {Total document size, frequent used word, average sentence length, frequency distribution of average sentence length}

Table 12: Set 12

Machine Learning	Accuracy	Precision	Recall	Fmeasure
NaiveBayes	54.651%	0.595	0.547	0.555
LibSVM	73.258%	0.772	0.733	0.738
MultiLayerPerceptron	62.791%	0.638	0.628	0.627
SMO	44.358%	0.638	0.628	0.627
DecissionStump	46.512%	0.239	0.465	0.311

Comparing table 11 and table 12, it is seen that accuracy is increased in each machine learning algorithm. If we use MultiLayerPerceptron algorithm, when frequently used word and total document size is considered, we get 47% accuracy. But incorporate it with average sentence length and its frequency distribution, we get 63% accuracy.

5.1.2 Syntactic Features

Set 13: {Average noun phrase}.

Table 13: Set 13

Machine Learning	Accuracy	Precision	Recall	Fmeasure
NaiveBayes	29.069%	0.194	0.291	0.221
LibSVM	39.535%	0.281	0.395	0.324
MultiLayerPerceptron	37.2093%	0.263	0.372	0.298
SMO	36.046%	0.293	0.36	0.272
DecissionStump	40.697%	0.194	0.407	0.26

Set 14: {Average noun phrase, frequency distribution of average noun phrase}.

Table 14: Set 14

Machine Learning	Accuracy	Precision	Recall	Fmeasure
NaiveBayes	32.558%	0.333	0.326	0.316
LibSVM	53.488%	0.503	0.535	0.507
MultiLayerPerceptron	51.063%	0.487	0.512	0.49
SMO	45.349%	0.235	0.453	0.302
DecissionStump	45.349%	0.239	0.453	0.304

Comparing table 13 and table 14 it is seen that accuracy is increased in each machine learning algorithm. If we use MultiLayerPerceptron algorithm, when only average noun phrase is considered, we get 37% accuracy. But incorporate it with its frequency distribution, we get 51% accuracy.

Set 15: {Average parsed tree height}

Table 15: Set 15

Machine Learning	Accuracy	Precision	Recall	Fmeasure
NaiveBayes	43.023%	0.338	0.43	0.372
LibSVM	48.8372%	0.43	0.488	0.452
MultiLayerPerceptron	48.837%	0.43	0.488	0.452
SMO	45.349%	0.235	0.453	0.302
DecissionStump	45.349%	0.239	0.453	0.304

Set 16: {Average parsed tree height, frequency distribution of average parsed tree height}

Table 16: Set 16

Machine Learning	Accuracy	Precision	Recall	Fmeasure
NaiveBayes	43.0239%	0.338	0.43	0.372
LibSVM	48.8372%	0.43	0.488	0.452
MultiLayerPerceptron	51.163%	0.487	0.512	0.49
SMO	53.489%	0.503	0.535	0.507
DecissionStump	45.349%	0.239	0.453	0.304

Comparing table 15 and table 16, it is seen that accuracy is increased in each machine learning algorithm. If we use average parsed tree height incorporate with its frequency distributions instead of only use average parsed tree height we can get better accuracy. If we use SMO algorithm, when only average parsed tree height is considered, we get 45% accuracy. But incorporate it with its frequency distribution, we get 54% accuracy.

5.1.3 Shallow Features Integrated with Syntactic Features

Set 17: {Total size, average sentence length, frequency distribution of average sentence length, average noun phrase, frequency distribution of average noun phrase}

Table 17: Set 17

Machine Learning	Accuracy	Precision	Recall	Fmeasure
NaiveBayes	51.163%	0.538	0.512	0.498
LibSVM	66.279%	0.736	0.663	0.665
MultiLayerPerceptron	68.605%	0.688	0.686	0.68
SMO	50%	0.438	0.5	0.462
DecissionStump	70.581%	0.239	0.465	0.311

Set 18: {Total size, average sentence length, frequency distribution of average sentence length, average noun phrase, frequency distribution of average noun phrase, largest noun phrase}

Table 18: Set 18

Machine Learning	Accuracy	Precision	Recall	Fmeasure
NaiveBayes	52.3256%	0.535	0.523	0.51
LibSVM	66.279%	0.736	0.663	0.665
MultiLayerPerceptron	76.7442%	0.723	0.721	0.719
SMO	53.651%	0.511	0.547	0.517
DecissionStump	72.093%	0.239	0.465	0.311

From the table 17 and 18, it is seen that if we add maximum length of noun phrase incorporate with noun phrase then we have found better achievement. If we use Neural Network (MultilayerPerceptron) machine learning algorithm we get better accuracy which is 76.44%.

5.2 Comparison between previous approach and proposed approach

We have shown the comparison between previous approach and proposed approach using graph. From the figure 1, it is seen that considering average sentence length with its frequency distribution value increases accuracy. Using SVM (RBF Kernel) accuracy increases to 36.046% to 38.372%. Using artificial neural network it increases to 39.535% to 41.035%.

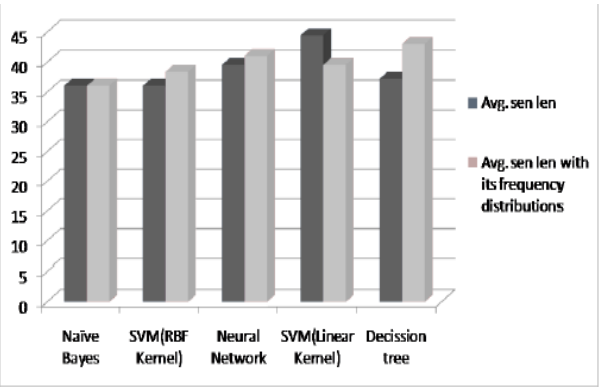

Fig. 1: {Average Sentence Length} VS {Average Sentence Length With its Frequency Distribution.}.

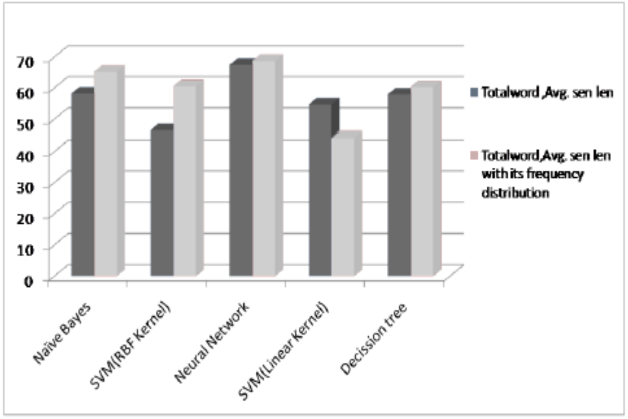

Fig. 2: {Total Word, Average Sentence Length} VS {Total Word, Average Sentence Length with its Frequency Distribution}.

From figure 2, it is seen that considering the set Total Word, Average Sentence Length VS Total Word, Average Sentence Length with its Frequency Distribution accuracy increases using SVM (RBF Kernel) 13.95% and using neural network 1.20% and using decision tree 2.33%. In table 4.2 using only average sentence length accuracy using Support Vector Machine 46.511% and using Neural Network 67.422%. From table 3, using average sentence length incorporate with its frequency distributed value using Support Vector Machine 60.465% and using Neural Network 68.605%.

In table 13 ,using only average noun phrase accuracy using Support Vector Machine 39.535% and using Neural Network 37.209%. But from table 14 using average noun phrase with its frequency distributed value accuracy using Support Vector Machine 41.861% and using Neural Network 44.186%. Using SVM (Linear Kernel) the value increases from 36. 047% to 40.698%. .Using decision tree these value increases from 46.515% to 48.837%. These situations are described in the figure 3.

During testing we have get some surprising result. If we incorporate largest verb phrase with verb phrase phrases and its average frequency distributed value then we have got better accuracy. Consider the input set {Total word, Average sentence length with its frequency distribution, Average Verb Phrase with its frequency distribution } and {Total word, Average sentence length with its frequency distribution, Average Verb Phrase with its frequency distribution, largest Verb Phrase}. The results have been given in figure 4.

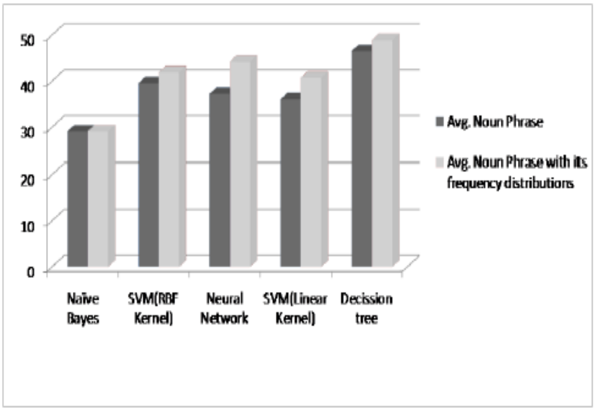

Fig. 3: {Average Noun Phrase} VS {Average Noun Phrase with its Frequency Distribution}.

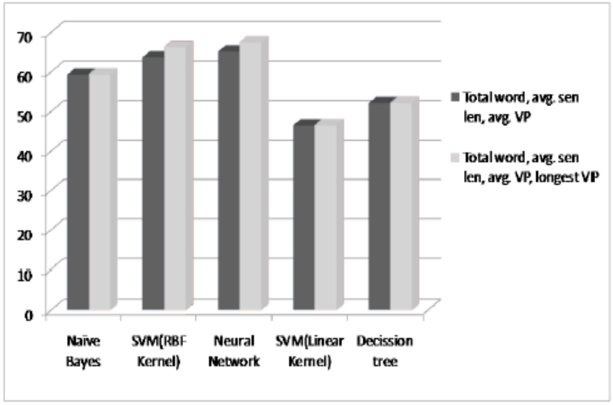

Fig. 4: {Total word , Average sentence length with its frequency distribution, Average Verb Phrase with its frequency distribution }VS {Total word , Average sentence length with its frequency distribution, Average Verb Phrase with its frequency distribution, largest Verb Phrase } .

From the figure 4, it is seen that using SVM (RBF kernel) accuracy increases 2.63% and using artificial neural network accuracy increases to 2.33%.

My best accuracy is 76.75% which i have got considering the {Total word, Average sentence length with its frequency distributions, Average Noun Phrase with its frequency distribution, largest Noun Phrase }and using artificial neural network algorithm. In previous approach, only {Total word, Average sentence length, Average Noun Phrase} was considered. The result of comparison between two sets is shown in the figure 5.

Using artificial neural network accuracy increases to 8.14% and using support vector machine accuracy increases to 3.49% and using decision tree accuracy increases to 1.51%.

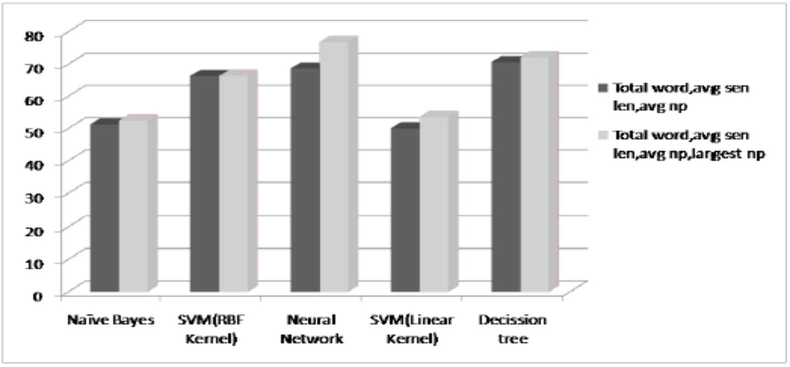

Fig. 5: {Total word, Average sentence length with its frequency distributions, Average Noun Phrase with its frequency distributions}VS {Total word , Average sentence length with its frequency distributions, Average Noun Phrase with its frequency distribution, largest Noun Phrase}.

6. Discussion and conclusion

In this research, we have improved an existing algorithm to determine the readability level of a text document. The aim of our research work is that using various fields of features adding or incorporating some new features how we can get better accuracy. In short, the research has two main results. First, in feature selection we have seen that if we use any average value incorporate with its frequency distributed value accuracy measure is increased. Second, if we use maximum noun phrase incorporate with average noun phrase and its frequency distributed value accuracy is 76%. This is our best consideration.

There are number of directions where improvement can take place in future studies. Currently**,** we only consider shallow features and syntactic features. In future I have a plan to do more research on shallow features and syntactic features. Since the ultimate goal of readability determination is to get better accuracy, this research can be extended on discourse features extraction especially semantic, co-reference and mental model dimensions metrics. We hope to improve our method to test our new feature on real annotated corpus and measure accuracy. Hope that using new feature we will get better accuracy. We also have an idea to build an automated software tools for readability determination. When some one wants to understand a topic from web resource can automatically test its difficulty level. For this condition, if we use discourse feature, it requires more time for computation because for extraction discourse feature from a text requires several scans on over all document. If we are able to increase our difficulty level, it will save time.

Readability assessment is not only useful to determine difficulty level of document but also necessary to have an automated system to produce a simplification version of that document. We have an idea to create a tool that determines a documents readability level and as well as produce a simplification version of that document. This tool can help someone to improve their reading and writing skill.

Acknowledgements

This research was supported by the Department of Computer Science & Engineering, University of Dhaka. We would like to thank the department for providing us lab facilities and research journals to complete the research.

References

[1] F. Sebastiani, "**Machine learning in automated text categorization**", ACM computing surveys (CSUR), vol. 34, no. 1, 2002, pp. 1-47.

[2] K. Collins-Thompson and J. Callan, "**A language modeling approach to predicting reading difficulty**" , in Proceedings of HLT/NAACL, 2004, vol. 4.

[3] A. Hart, "**Using neural networks for classification tasks{some experiments on datasets and practical advice**", The Journal of the Operational Research Society, vol. 43, no. 3, 1992, pp. 215-226.

[4]H. Isozaki, "**Japanese named entity recognition based on a simple rule generator and decision tree learning**", in Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, 2001, pp. 314-321.

[5]M. Bevilacqua, M. Braglia, and R. Montanari, "**The classification and regression tree approach to pump failure rate analysis**", Reliability Engineering & System Safety, vol. 79, no. 1, 2003, pp. 59-67.

[6]D. D'Alessandro, P. Kingsley, and J. Johnson-West, "**The readability of pediatric patient education materials on the World Wide Web**," Archives of pediatrics and adolescent medicine, vol. 155, no. 7,2001, p.807.

[7]L. Feng, M. Jansche, M. Huenerfauth, and N. Elhadad, "**A Comparison of Features for Automatic Readability Assessment**", in 23rd International Conference on Computational Linguistics (COLING 2010), pp. 276-284.

[8]M. Heilman, L. Zhao, J. Pino, and M. Eskenazi, "**Retrieval of reading materials for vocabulary and reading practice**", in Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications, 2008, pp. 80-88.

[9]J. Chae and A. Nenkova, "**Predicting the fluency of text with shallow structural features: case studies of machine translation and human-written text**", in Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, 2009, pp. 139-147